ECOSORT

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

S. Yaswanth Sai	**(22501A05F8)**
P. Sree Satya Harsha	**(22501A05E4)**
S. Surya Harsha	**(22501A05F7)**
SK. Irfan Haris	**(22501A05G2)**

Under the Guidance of

Mr. Michael Sadgun Rao Kona,

Assistant Professor

PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY

(Permanently affiliated to JNTU : Kakinada, Approved by AICTE)

(An NBA & NAAC A+ accredited and ISO 9001:2015 Certified Institution)

Kanuru, Vijayawada - 520007

2024-25

PRASAD V POTLURI

SIDDHARTHA INSTITUTE OF TECHNOLOGY

(Permanently affiliated to JNTU : Kakinada, Approved by AICTE)

(An NBA & NAAC A+ accredited and ISO 9001:2015 certified institution)

Kanuru, Vijayawada – 520007

CERTIFICATE

This is to certify that the project report title **"ECOSORT"** is the bonafied work of **S. Yaswanth (22501A05F8), P. Sree Satya Harsha (22501A05E4), S. Surya Harsha (22501A05F7), SK. Irfan Haris (22501A05G2)** in partial fulfilment of completing Academic project in Mobile App Development (20SA8651) during academic year 2024-25.

Signature of the Incharge **Signature of the HOD**

INDEX

1. ABSTRACT

EcoSort is an innovative Android application designed to promote awareness about waste segregation through community-driven content. Built using Java for Android and Firebase as the backend, the app ensures a seamless user experience with real-time data management and secure authentication.
The system classifies users into two roles:
Activists – Individuals who contribute educational content on waste segregation.
Normal Users – Users who can access and learn from the content shared by activists.

During registration, users specify their role, which determines their home page functionality. Activists have access to an interactive interface where they can post descriptions to educate others, while normal users can only view the shared content. The application integrates user authentication and secure data storage using Firebase, ensuring personalized user experiences.

A key feature of EcoSort is its intuitive user interface that allows activists to seamlessly add content with a simple tap of a button. The home page dynamically updates with new content, keeping users engaged and informed. By leveraging Firebase, the app ensures efficient data retrieval, allowing real-time content updates. Eco Sort serves as a digital platform for environmental education, encouraging responsible waste disposal and sustainability practices. By fostering knowledge-sharing and community engagement, it empowers individuals to take proactive steps toward a cleaner and greener future. The app bridges the gap between awareness and action, ensuring that waste management education reaches a wider audience and contributes to long-term environmental benefits.

1.1. SDG JUSTIFICATION REPORT

1.SDG 3: Good Health & Well-being

This goal focuses on ensuring **healthy lives and promoting well-being** for all. EcoSort contributes to SDG 3 by improving waste disposal practices, reducing pollution, and minimizing health risks caused by improper waste management.

How EcoSort Supports SDG 3:

- **Minimizing Health Hazards**: Reduces exposure to hazardous waste and prevents health risks from improper disposal.
- **Promoting Clean Environments:** Encourages proper waste segregation, reducing landfill overflow and improving air quality.
- **Awareness & Education:** Educates users on the impact of waste on public health and provides tips for responsible disposal.

2. SDG 6: Clean Water & Sanitation

This goal aims to ensure the availability and sustainable management of water resources. Ecosort helps prevent water contamination caused by improper waste disposal, contributing to **cleaner water bodies and better sanitation practices**.

How Ecosort Supports SDG 6:

- **Preventing Water Pollution:** Encourages proper disposal of non-biodegradable and toxic waste, reducing contamination of rivers and groundwater.
- **Sustainable Waste Management:** Helps users identify recyclable and hazardous waste, ensuring it is processed correctly to prevent harmful runoff.
- **Education on Safe Disposal:** Raises awareness about how improper disposal of plastics, chemicals, and medical waste affects water quality.

3. SDG 12: Responsible Consumption & Production

Ecosort aligns with SDG 12 by encouraging individuals and communities to adopt sustainable consumption habits and reduce waste generation**.**

How EcoSort Supports SDG 12:

- **Promoting Responsible Waste Disposal:** Helps users sort and dispose of waste efficiently, minimizing landfill overflow.
- **Encouraging Recycling & Upcycling:** Provides recommendations on how to repurpose or recycle items instead of discarding them.
- **Data-Driven Waste Tracking:** Allows users to monitor and analyze their waste habits, encouraging reduction in waste production.
- **Community Engagement:** Connects users with local recycling initiatives, inspiring collective action for sustainable waste management.

4. SDG 13: Climate Action

This goal emphasizes urgent action to combat climate change and its impacts. EcoSort helps mitigate climate change by reducing carbon emissions linked to waste mismanagement.

How Ecosort Supports SDG 13:

- **Reducing Greenhouse Gas Emissions:** Encourages composting and proper waste disposal to lower methane emissions from landfills.
- **Promoting Sustainable Alternatives:** Educates users on reducing plastic waste and switching to eco-friendly products.
- **Encouraging Circular Economy Practices:** Advocates for recycling and upcycling to decrease waste production and resource depletion**.**

5. SDG 15: Life on Land

Ecosort plays a crucial role in preserving terrestrial ecosystems by preventing land degradation and protecting biodiversity.

How Ecosort Supports SDG 15:

- **Reducing Land Pollution:** Promotes proper waste segregation, preventing harmful materials from contaminating soil.
- **Protecting Wildlife & Ecosystems:** Ensures hazardous waste is disposed of safely, preventing harm to animals and natural habitats.
- **Advocating for Sustainable Practices:** Encourages users to adopt eco-friendly habits that support long-term environmental conservation.

2. INTRODUCTION

EcoSort is an innovative mobile application designed to educate and empower individuals in effective waste segregation. As improper waste disposal continues to pose a significant environmental challenge, Eco Sort provides a structured platform where users can learn best practices for managing waste sustainably. The app categorizes users into two roles: **Activists**, who contribute educational content, and **Normal Users**, who access and learn from these contributions. This ensures that waste management knowledge is shared effectively within the community.

With a user-friendly interface, **EcoSort** allows activists to post insightful descriptions about waste segregation, recycling methods, and eco-friendly disposal techniques. Normal users, in turn, can browse and adopt these sustainable habits, fostering an environmentally responsible society. The app seamlessly integrates Firebase for secure user authentication and real-time content updates, ensuring a smooth and engaging experience.

Beyond simple content sharing, **EcoSort** promotes awareness campaigns, intelligent waste sorting tips, and community-driven discussions on sustainability. By leveraging technology to bridge the gap between awareness and action, **EcoSort** plays a crucial role in reducing waste pollution, conserving resources, and creating a cleaner, greener future.

3. OBJECTIVES AND SCOPE OF THE PROJECT

Objectives of the project:

1. **Promote Waste Segregation Awareness** – Educate users about the importance of proper waste disposal and encourage environmentally responsible habits through activist-generated content.
2. **Provide a Knowledge-Sharing Platform** – Allow activists to share valuable insights and awareness messages about waste segregation, recycling, and sustainable waste management.
3. **Facilitate Role-Based User Experience** – Ensure a structured interaction where activists can add descriptions, and normal users can access and learn from the content.
4. **Ensure Secure User Authentication** – Utilize Firebase for secure user registration and login, ensuring personalized experiences based on user roles.
5. **Encourage Community Participation** – Foster a collaborative environment where users engage with eco-friendly practices, spreading awareness about responsible waste management.
6. **Enhance Accessibility and Engagement** – Design a user-friendly interface that allows users to navigate content easily, ensuring effective learning and adoption of waste segregation habits.

Scope of the Project:

1. **User Registration & Authentication:** Users must register with their details and choose between "Activist" or "Normal User" roles to access the app's functionalities.
2. **Activist Home Page:** Activists can add descriptions and awareness messages related to waste segregation.
3. **User Home Page:** Normal users can view and interact with content shared by activists to learn and implement sustainable waste management practices.
4. **Content Management:** Activists can only add descriptions, ensuring streamlined content creation without unnecessary modifications.
5. **Real-time Data Storage:** Firebase integration ensures efficient data storage, retrieval, and updates for activist-generated content.
6. **User Engagement & Motivation:** By providing easy access to informative content, the app encourages users to adopt better waste management practices.

4. SOFTWARE USED – EXPLANATION

The **ECOSORT** application is developed using a combination of **Android, Java, Fire Base** to create a seamless and efficient agricultural advisory system.

Android

Android is an open-source mobile operating system developed by Google, designed primarily for touchscreen devices. It provides a powerful platform for creating interactive, user-centric mobile applications.

Usage in ECOSORT

- The ECOSORT application is built entirely on the Android platform, making it accessible to users via their smartphones.
- Android's flexible UI framework using XML is utilized for designing clean, responsive, and user-friendly interfaces.
- Core Android components such as **Activities, Fragments, RecyclerView, and ViewPager** are used to ensure a seamless user experience.
- Android’s notification and lifecycle management features ensure smooth app functionality, even during background tasks like data retrieval and usage tracking.

Java

Java is a versatile, object-oriented programming language that provides stability, reliability, and platform independence, making it a primary choice for Android app development.

Usage in ECOSORT

- It manages **user authentication and content storage** using **Firebase Firestore**, ensuring activists can add descriptions while users can view them in real time.
- **Java’s exception handling** ensures smooth app operation, preventing crashes due to invalid inputs or unexpected errors.
- The application utilizes Java to implement **role-based access control**, allowing activists to post descriptions while restricting normal users to viewing content.

Fire Base

Usage in Eco Sort

- **Cloud Firestore as Database:** Firebase Firestore is used as the primary database to store user details, roles (activist or user), and the descriptions added by activists.
- **User Authentication:** Firebase Authentication ensures secure login and registration for both activists and users, enabling role-based access.
- **Real-time Updates:** Firestore enables real-time synchronization, ensuring that users can instantly view the latest descriptions posted by activists.
- **Data Security:** Firebase provides built-in security rules, preventing unauthorized modifications and ensuring only activists can add content while normal users can view it.

Development Tools

Code Editor: Android Studio is the official IDE for Android development, providing a powerful code editor, visual layout editor, and debugging tools. It supports Java and integrates seamlessly with Firebase, making it an ideal choice for developing feature-rich mobile applications. The IDE includes an emulator for testing applications across different devices and screen sizes, ensuring a smooth user experience. Android Studio also offers real-time error detection and intelligent code completion, helping developers write optimized and bug-free code.

Version Control System (VCS): GitHub is a widely used version control system that helps manage project files, track changes, and collaborate with team members. It enables developers to work on different features simultaneously using branches and ensures that all modifications are safely stored and documented. By utilizing GitHub, the project remains organized, with a history of code changes that can be reviewed and reverted if needed. This improves code stability and allows multiple developers to contribute efficiently.

5. PROPOSED MODEL

1. Overview

The EcoSort application is a mobile-based solution designed to educate users on effective waste segregation practices. It serves two types of users: Activists, who can add and share educational content, and Normal Users, who can browse and learn about sustainable waste disposal methods. The application integrates user authentication, real-time content updates, and a simple, user-friendly interface to promote environmental awareness and responsible waste management.

2. System Components

User Authentication and Profile Management

- Users can register and log in securely, selecting their role as either an Activist or a Normal User.
- The profile section stores essential user details such as name, email, and role.
- Activists have the ability to post descriptions related to waste segregation.
- Normal Users can only view the content shared by activists.

Content Posting and Browsing Module

- **Activists** can add educational descriptions about waste segregation and proper disposal methods.
- **Normal Users** can browse and read the content to gain insights into sustainable waste management.

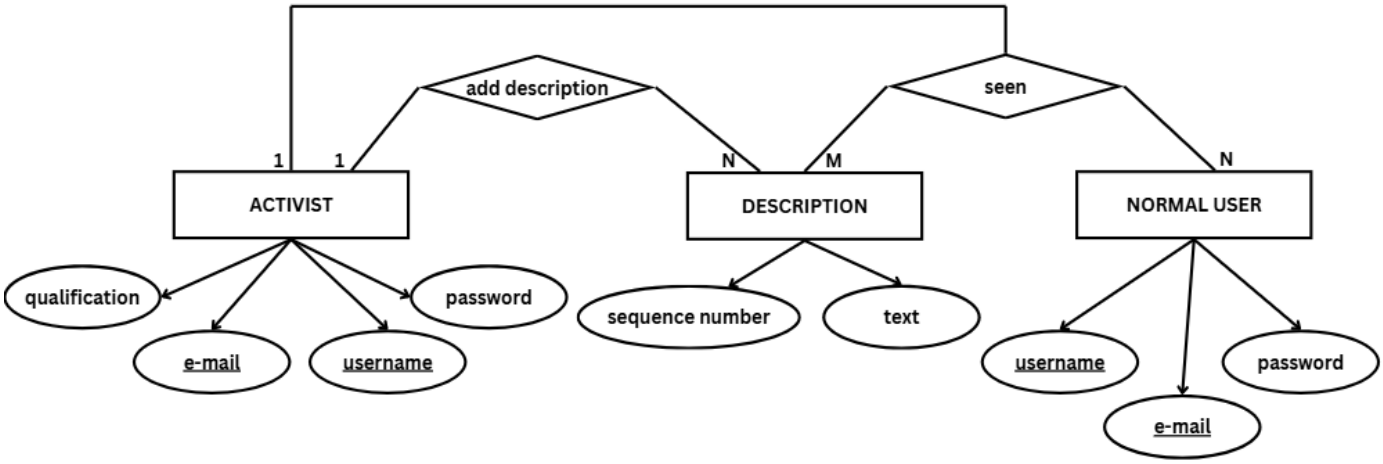
Data Storage and Security

- Firebase Authentication ensures secure user login and role-based access control.
- Firestore manages and retrieves educational content efficiently while maintaining data security.

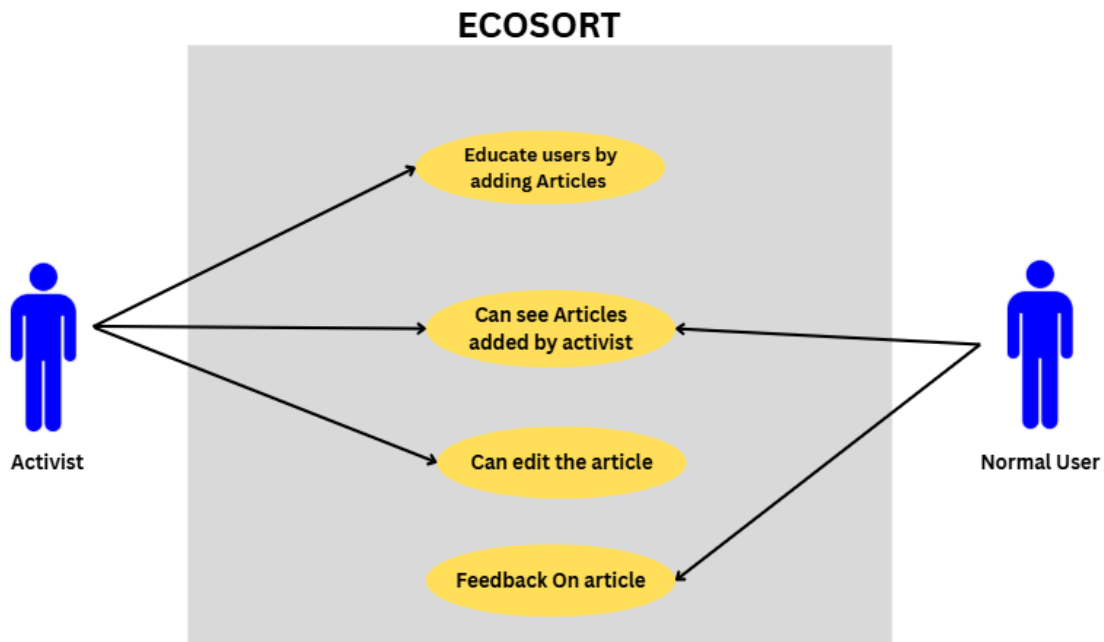
3. Entity-Relationship Model

The database is structured with key entities:

- **User to Description** → (1 to Many)
- **Activist to Added Descriptions** → (1 to Many)
- **Normal User to Viewed Descriptions** → (Many to Many)



Use Case Diagram:



4. System Workflow

1. The user (normal user or activist) registers/logs into the EcoSort application.
2. The system retrieves previous activist posts stored in the application for users to view.
3. The normal user can only view the shared environmental experiences and insights.
4. The activist has the ability to add new posts, sharing their knowledge, experiences, and awareness messages.
5. The app displays all activist contributions in a structured format for users to engage with.

5. Future Enhancements

The system can be expanded to include additional features such as:

1. **AI-Driven Environmental Insights** – Integration of AI to analyze user-generated content, detect emerging environmental issues, and provide personalized recommendations for sustainable practices.
2. **Gamification and Community Challenges** – Introducing rewards, badges, and eco-challenges to encourage user participation and foster a community-driven approach to sustainability.
3. **Crowdsourced Environmental Reporting** – Enabling users to report environmental concerns such as pollution, deforestation, and waste mismanagement, which can be mapped and shared with relevant organizations.
4. **Real-Time Environmental Data Integration** – Connecting with open environmental data sources to provide real-time updates on air quality, climate conditions, and sustainability trends.

6. SAMPLE CODE

Github repository link: https://github.com/yaswan123/Ecosort

Folder Structure:

```
ECOSORT
 +---app
 |  +---main
 |  |  |  AndroidManifest.xml
 |  |  |
 |  |  +---java
 |  |  |  \---com
 |  |  |      \---example
 |  |  |          \---ecosort
 |  |  |              Activist.java
 |  |  |              ActivistHomePage.java
 |  |  |              ActivistLoginActivity.java
 |  |  |              AddNotes.java
 |  |  |              Home.java
 |  |  |              LoginActivity.java
 |  |  |              MainActivity.java
 |  |  |              NoteEntity.java
 |  |  |              NotesAdapter.java
 |  |  |              RegisterActivity.java
 |  |  |              UserProfile.java
 |  |  |              Users.java
 |  |  |
 |  |  \---res
 |  |      +---drawable
 |  |      |
 |  |      +---layout
 |  |      |  activity_activist_home_page.xml
 |  |      |  activity_activist_login_page.xml
 |  |      |  activity_add_notes.xml
 |  |      |  activity_home.xml
 |  |      |  activity_login.xml
 |  |      |  activity_main.xml
 |  |      |  activity_register.xml
 |  |      |  activity_userprofile.xml
 |  |      |  note_item.xml
 |  |      |
 |  |      +---mipmap
 |  |      +---raw
 |  |      +---values
 |  |      +---xml
 |  |
 |  \---res (generated)
```

MainActivity.java

```
package com.example.ecosort;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private TextView registerText, loginText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        registerText = findViewById(R.id.registertext);
        loginText = findViewById(R.id.logintext);
        registerText.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this, RegisterActivity.class);
            startActivity(intent);
        });
        loginText.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this, LoginActivity.class);
            startActivity(intent);
        });
    }
}
```

ActivistHomePage.java

```
package com.example.ecosort;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.google.firebase.firestore.EventListener;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;
import com.google.firebase.firestore.QuerySnapshot;
import java.util.ArrayList;
import java.util.List;
public class ActivistHomePage extends AppCompatActivity {
    private RecyclerView recyclerView;
    private NotesAdapter adapter;
    private FirebaseFirestore db;
    private List<NoteEntity> noteList;
    private ImageButton addNotesButton;
```

```
    private Button logoutButton; // Logout button
    private SharedPreferences sharedPreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_activist_home_page);
        sharedPreferences = getSharedPreferences("UserSession", Context.MODE_PRIVATE);
        // Ensure user is logged in
        if (!sharedPreferences.getBoolean("isLoggedIn", false)) {
            Log.e("ActivistHomePage", "User not logged in. Redirecting to login.");
            startActivity(new Intent(this, ActivistLoginActivity.class));
            finish();
            return;
        }
        // Initialize UI components
        recyclerView = findViewById(R.id.recycler_view);
        addNotesButton = findViewById(R.id.addnotes);
        logoutButton = findViewById(R.id.logoutButton); // Ensure this exists in XML
        if (logoutButton == null) {
            Log.e("ActivistHomePage", "Logout button is NULL! Check XML layout.");
        }
        db = FirebaseFirestore.getInstance();
        noteList = new ArrayList<>();
        adapter = new NotesAdapter(noteList);
        recyclerView.setAdapter(adapter);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        loadNotes();
        addNotesButton.setOnClickListener(v -> startActivity(new Intent(this, AddNotes.class)));
        logoutButton.setOnClickListener(v -> logoutUser()); // Logout functionality
    }

    private void loadNotes() {
        db.collection("notes").addSnapshotListener(new EventListener<QuerySnapshot>() {
            @Override
            public void onEvent(QuerySnapshot value, FirebaseFirestoreException error) {
                if (error != null) {
                    Toast.makeText(ActivistHomePage.this, "Error loading notes!",
Toast.LENGTH_SHORT).show();
                    return;
                }
                noteList.clear();
                if (value != null) {
                    for (var doc : value.getDocuments()) {
                        NoteEntity note = doc.toObject(NoteEntity.class);
                        noteList.add(note);
                    }
                }
                adapter.notifyDataSetChanged();
            }
        });
    }

    private void logoutUser() {
        // Clear session
```

```
        sharedPreferences.edit().putBoolean("isLoggedIn", false).apply();

        Toast.makeText(this, "Logged out successfully!", Toast.LENGTH_SHORT).show();
        startActivity(new Intent(this, ActivistLoginActivity.class));
        finish();
    }
}
```

ActivistLoginActivity.java

```
package com.example.ecosort;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
public class ActivistLoginActivity extends AppCompatActivity {
    private static final String TAG = "ActivistLoginActivity";
    private EditText emailField, passwordField;
    private Button loginButton;
    private DatabaseReference activistsRef;
    private SharedPreferences sharedPreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_activist_login_page);
        emailField = findViewById(R.id.username);
        passwordField = findViewById(R.id.userpassword);
        loginButton = findViewById(R.id.loginbuttton);

        activistsRef = FirebaseDatabase.getInstance().getReference("activists");
        sharedPreferences = getSharedPreferences("UserSession", Context.MODE_PRIVATE);

        // Auto-login if already logged in
        if (sharedPreferences.getBoolean("isLoggedIn", false)) {
            startActivity(new Intent(this, ActivistHomePage.class));
            finish();
        }
        loginButton.setOnClickListener(v -> checkActivistLogin());
    }
    private void checkActivistLogin() {
        String emailInput = emailField.getText().toString().trim();
        String passwordInput = passwordField.getText().toString().trim();

        if (emailInput.isEmpty() || passwordInput.isEmpty()) {
```

```
            Toast.makeText(this, "Enter email and password!", Toast.LENGTH_SHORT).show();
            return;
        }
        String emailKey = emailInput.replace(".", "_").replace("#", "_").replace("$", "_")
                .replace("[", "_").replace("]", "_");

        activistsRef.child(emailKey).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    Activist activist = snapshot.getValue(Activist.class);
                    if (activist != null && activist.getPassword().equals(passwordInput)) {
                        // Save login session
                        sharedPreferences.edit().putBoolean("isLoggedIn", true).apply();

                        Toast.makeText(ActivistLoginActivity.this, "Login Successful!",
Toast.LENGTH_SHORT).show();
                        startActivity(new Intent(ActivistLoginActivity.this, ActivistHomePage.class));
                        finish();
                    } else {
                        Toast.makeText(ActivistLoginActivity.this, "Invalid password!",
Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(ActivistLoginActivity.this, "Activist not found!",
Toast.LENGTH_SHORT).show();
                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(ActivistLoginActivity.this, "Database error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

AddNotes.java

```
package com.example.ecosort;
import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.firestore.FirebaseFirestore;
import java.util.HashMap;
import java.util.Map;
public class AddNotes extends AppCompatActivity {
    private EditText notesInput;
    private Button saveButton;
    private FirebaseFirestore db;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_notes);
        notesInput = findViewById(R.id.notes);
        saveButton = findViewById(R.id.addnote);
        db = FirebaseFirestore.getInstance();
        saveButton.setOnClickListener(v -> saveNote());
    }
    private void saveNote() {
        String userNote = notesInput.getText().toString().trim();
        if (userNote.isEmpty()) {
            Toast.makeText(this, "Please enter a note!", Toast.LENGTH_SHORT).show();
            return;
        }
        SharedPreferences sharedPreferences = getSharedPreferences("UserSession",
Context.MODE_PRIVATE);
        String username = sharedPreferences.getString("username", "Unknown User");
        String noteId = db.collection("notes").document().getId();
        Map<String, Object> noteData = new HashMap<>();
        noteData.put("noteId", noteId);
        noteData.put("noteContent", userNote);
        noteData.put("username", username);
        db.collection("notes").document(noteId).set(noteData)
                .addOnSuccessListener(aVoid -> {
                    Toast.makeText(this, "Note added successfully!", Toast.LENGTH_SHORT).show();
                    finish();
                })
                .addOnFailureListener(e -> Toast.makeText(this, "Failed to add note!",
Toast.LENGTH_SHORT).show());
    }
}
```

Home.java

```
package com.example.ecosort;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.google.firebase.firestore.EventListener;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;
import com.google.firebase.firestore.QuerySnapshot;
import java.util.ArrayList;
import java.util.List;
public class Home extends AppCompatActivity {
    private RecyclerView recyclerView;
```

```
private NotesAdapter adapter;
private FirebaseFirestore db;
private List<NoteEntity> noteList;
private Button logoutButton; // Logout button
private SharedPreferences sharedPreferences;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home);
    sharedPreferences = getSharedPreferences("UserSession", Context.MODE_PRIVATE);
    // Ensure user is logged in
    if (!sharedPreferences.getBoolean("isLoggedIn", false)) {
        Log.e("Home", "User not logged in. Redirecting to login.");
        startActivity(new Intent(this, LoginActivity.class)); // Ensure this activity exists
        finish();
        return;
    }
    // Initialize UI components
    recyclerView = findViewById(R.id.recycler_view);
    logoutButton = findViewById(R.id.logoutButton);

    if (logoutButton == null) {
        Log.e("Home", "Logout button is NULL! Check XML layout.");
    }
    db = FirebaseFirestore.getInstance();
    noteList = new ArrayList<>();
    adapter = new NotesAdapter(noteList);
    recyclerView.setAdapter(adapter);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    loadNotes();
    logoutButton.setOnClickListener(v -> logoutUser());
}
private void loadNotes() {
    db.collection("notes").addSnapshotListener(new EventListener<QuerySnapshot>() {
        @Override
        public void onEvent(QuerySnapshot value, FirebaseFirestoreException error) {
            if (error != null) {
                Toast.makeText(Home.this, "Error loading notes!", Toast.LENGTH_SHORT).show();
                return;
            }
            noteList.clear();
            if (value != null) {
                for (var doc : value.getDocuments()) {
                    NoteEntity note = doc.toObject(NoteEntity.class);
                    noteList.add(note);
                }
            }
            adapter.notifyDataSetChanged();
        }
    });
}
private void logoutUser() {
    // Clear session
    sharedPreferences.edit().putBoolean("isLoggedIn", false).apply();
```

```
        Toast.makeText(this, "Logged out successfully!", Toast.LENGTH_SHORT).show();
        startActivity(new Intent(this, LoginActivity.class)); // Ensure this activity exists
        finish();
    }
}
```

LoginActivity.java

```
package com.example.ecosort;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
public class LoginActivity extends AppCompatActivity {
    private EditText emailField, passwordField;
    private Button loginButton;
    private TextView activistLogin; // Activist text
    private DatabaseReference usersRef;
    private SharedPreferences sharedPreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        emailField = findViewById(R.id.username);
        passwordField = findViewById(R.id.userpassword);
        loginButton = findViewById(R.id.userlogin);
        activistLogin = findViewById(R.id.popup); // Get Activist TextView
        usersRef = FirebaseDatabase.getInstance().getReference("users");
        sharedPreferences = getSharedPreferences("UserSession", Context.MODE_PRIVATE);
        // Auto-login if already logged in
        if (sharedPreferences.getBoolean("isLoggedIn", false)) {
            startActivity(new Intent(this, Home.class)); // Redirect to Home page
            finish();
        }
        loginButton.setOnClickListener(v -> checkUserLogin());
        // Set click listener for activist login
        activistLogin.setOnClickListener(v -> {
            Intent intent = new Intent(LoginActivity.this, ActivistLoginActivity.class);
            startActivity(intent);
        });
    }
```

```
    private void checkUserLogin() {
        String emailInput = emailField.getText().toString().trim();
        String passwordInput = passwordField.getText().toString().trim();
        if (emailInput.isEmpty() || passwordInput.isEmpty()) {
            Toast.makeText(this, "Enter email and password!", Toast.LENGTH_SHORT).show();
            return;
        }
        String emailKey = emailInput.replace(".", "_").replace("#", "_").replace("$", "_")
                .replace("[", "_").replace("]", "_");
        usersRef.child(emailKey).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    Users user = snapshot.getValue(Users.class);
                    if (user != null && user.getPassword().equals(passwordInput)) {
                        // Save login session
                        sharedPreferences.edit().putBoolean("isLoggedIn", true).apply();
                        Toast.makeText(LoginActivity.this, "Login Successful!",
Toast.LENGTH_SHORT).show();
                        startActivity(new Intent(LoginActivity.this, Home.class)); // Redirect to Home page
                        finish();
                    } else {
                        Toast.makeText(LoginActivity.this, "Invalid password!",
Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(LoginActivity.this, "User not found!", Toast.LENGTH_SHORT).show();
                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(LoginActivity.this, "Database error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

NoteEntity.java

```
package com.example.ecosort;
public class NoteEntity {
    private String noteId;
    private String noteContent;
    private String username;
    public NoteEntity() {}
    public NoteEntity(String noteId, String noteContent, String username) {
        this.noteId = noteId;
        this.noteContent = noteContent;
        this.username = username;
    }
    // Getters
    public String getNoteId() {
        return noteId;
    }
```

```
    public String getNoteContent() {
        return noteContent;
    }
    public String getUsername() {
        return username;
    }
    // Setters (if needed)
    public void setNoteId(String noteId) {
        this.noteId = noteId;
    }
    public void setNoteContent(String noteContent) {
        this.noteContent = noteContent;
    }
    public void setUsername(String username) {
        this.username = username;
    }
}
```

NotesAdapter.java

```
package com.example.ecosort;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;
public class NotesAdapter extends RecyclerView.Adapter<NotesAdapter.NoteViewHolder> {
    private List<NoteEntity> noteList;
    public NotesAdapter(List<NoteEntity> noteList) {
        this.noteList = noteList;
    }
    @NonNull
    @Override
    public NoteViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.note_item, parent, false);
        return new NoteViewHolder(view);
    }
    @Override
    public void onBindViewHolder(@NonNull NoteViewHolder holder, int position) {
        NoteEntity note = noteList.get(position);
        // Remove the '#' from numbering
        holder.seriesNumber.setText(String.valueOf(position + 1));
        holder.noteContent.setText(note.getNoteContent());
    }
    @Override
    public int getItemCount() {
        return noteList.size();
    }
    public static class NoteViewHolder extends RecyclerView.ViewHolder {
        TextView seriesNumber, noteContent;
        public NoteViewHolder(View itemView) {
            super(itemView);
```

```
            seriesNumber = itemView.findViewById(R.id.note_series_number);
            noteContent = itemView.findViewById(R.id.note_content);
        }
    }
}
```

Users.java

```
package com.example.ecosort;
public class Users {
    private String username;
    private String email;
    private String role;
    private String qualification;
    private String password; // New field
    public Users() {
    }
    public Users(String username, String email, String role, String qualification, String password) {
        this.username = username;
        this.email = email;
        this.role = role;
        this.qualification = qualification;
        this.password = password;
    }
    // Getters
    public String getUsername() {
        return username;
    }
    public String getEmail() {
        return email;
    }
    public String getRole() {
        return role;
    }
    public String getQualification() {
        return qualification;
    }
    public String getPassword() {
        return password;
    }
}
}
```

RegisterActivity.java

```
package com.example.ecosort;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
```

```
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.database.DataSnapshot;

public class RegisterActivity extends AppCompatActivity {

    private EditText email, username, password, role, qualification;
    private Button registerButton;
    private DatabaseReference activistsRef, usersRef;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        email = findViewById(R.id.registeremail);
        username = findViewById(R.id.registerusername);
        password = findViewById(R.id.registerpassword);
        role = findViewById(R.id.registerrole);
        qualification = findViewById(R.id.registerqualification);
        registerButton = findViewById(R.id.registerbutton);
        FirebaseDatabase fdb = FirebaseDatabase.getInstance();
        activistsRef = fdb.getReference("activists");
        usersRef = fdb.getReference("users");
        registerButton.setOnClickListener(v -> insertUser());
    }
    private void insertUser() {
        String emailInput = email.getText().toString().trim();
        String usernameInput = username.getText().toString().trim();
        String passwordInput = password.getText().toString().trim();
        String roleInput = role.getText().toString().trim();
        String qualificationInput = qualification.getText().toString().trim();
        if (emailInput.isEmpty() || usernameInput.isEmpty() || passwordInput.isEmpty() ||
                roleInput.isEmpty() || qualificationInput.isEmpty()) {
            Toast.makeText(RegisterActivity.this, "All fields are required!",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (!android.util.Patterns.EMAIL_ADDRESS.matcher(emailInput).matches()) {
            Toast.makeText(RegisterActivity.this, "Invalid email format!",
Toast.LENGTH_SHORT).show();
            return;
        }
        String emailKey = emailInput.replace(".", "_").replace("#", "_")
                .replace("$", "_").replace("[", "_").replace("]", "_");
        DatabaseReference targetRef;
        boolean isActivist = roleInput.equalsIgnoreCase("activist");
        targetRef = isActivist ? activistsRef : usersRef;
        targetRef.child(emailKey).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    Toast.makeText(RegisterActivity.this, "User already registered!",
Toast.LENGTH_SHORT).show();
                } else {
                    if (isActivist) {
```

```
                Activist activist = new Activist(usernameInput, emailInput, roleInput,
qualificationInput, passwordInput);
                targetRef.child(emailKey).setValue(activist)
                    .addOnSuccessListener(unused -> {
                        Toast.makeText(RegisterActivity.this, "Activist Registered Successfully!",
Toast.LENGTH_SHORT).show();
                    })
                    .addOnFailureListener(e ->
                            Toast.makeText(RegisterActivity.this, "Error: " + e.getMessage(),
Toast.LENGTH_SHORT).show());
            } else {
                Users user = new Users(usernameInput, emailInput, roleInput, qualificationInput,
passwordInput);
                targetRef.child(emailKey).setValue(user)
                    .addOnSuccessListener(unused -> {
                        Toast.makeText(RegisterActivity.this, "User Registered Successfully!",
Toast.LENGTH_SHORT).show();
                    })
                    .addOnFailureListener(e ->
                            Toast.makeText(RegisterActivity.this, "Error: " + e.getMessage(),
Toast.LENGTH_SHORT).show());
            }
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        Toast.makeText(RegisterActivity.this, "Database error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
    }
  });
 }
}
```

activity_activist_home_page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/main"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".ActivistHomePage">
  <ImageView
    android:id="@+id/navbar"
    android:layout_width="match_parent"
    android:layout_height="70dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:srcCompat="@drawable/nav_bar" />
  <TextView
    android:id="@+id/title"
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="EcoSort"
        android:textSize="32sp"
        android:textStyle="bold"
        android:textAlignment="center"
        android:textColor="@color/black"
        app:layout_constraintTop_toTopOf="@+id/navbar"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:padding="10dp"
        app:layout_constraintTop_toBottomOf="@+id/navbar"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/buttons_layout" />
    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/buttons_layout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent">
        <Button
            android:id="@+id/logoutButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Logout"
            android:textColor="@color/white"
            android:backgroundTint="@color/black"
            android:padding="10dp"
            android:textSize="16sp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toStartOf="@+id/addnotes"
            app:layout_constraintHorizontal_bias="0.1" />
        <ImageButton
            android:id="@+id/addnotes"
            android:layout_width="70dp"
            android:layout_height="70dp"
            android:layout_margin="16dp"
            android:background="@null"
            android:clickable="true"
            android:focusable="true"
            android:scaleType="centerInside"
            android:onClick="changetoaddnotes"
            android:src="@drawable/baseline_add_circle_24"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_activist_login_page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/activistloginpage"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:background="@drawable/img"
    android:gravity="center"
    android:padding="20dp">
    <androidx.cardview.widget.CardView
        android:layout_width="365dp"
        android:layout_height="318dp"
        app:cardCornerRadius="30dp">
        <ImageView
            android:id="@+id/image"
            android:layout_width="match_parent"
            android:layout_height="67dp"
            app:srcCompat="@drawable/topbox" />
        <TextView
            android:id="@+id/logintxt"
            android:layout_width="match_parent"
            android:layout_height="65dp"
            android:text="LOGIN"
            android:textSize="24sp"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:textAlignment="center"
            android:layout_marginTop="13sp"
            android:padding="8dp"/>
        <LinearLayout
            android:layout_width="363dp"
            android:layout_height="324dp"
            android:layout_gravity="center_horizontal"
            android:orientation="vertical">
            <EditText
                android:id="@+id/username"
                android:layout_width="270dp"
                android:layout_height="55dp"
                android:layout_gravity="center"
                android:layout_marginTop="90sp"
                android:background="@drawable/txtbox"
                android:drawableLeft="@drawable/baseline_email_24"
                android:hint="G-MAIL"
                android:inputType="text"
                android:padding="8sp"
                android:paddingLeft="15sp"
                android:textColor="@color/material_dynamic_neutral30" />
            <EditText
                android:id="@+id/userpassword"
                android:layout_width="270dp"
                android:layout_height="55dp"
```

```
            android:layout_gravity="center"
            android:layout_marginTop="15sp"
            android:background="@drawable/txtbox"
            android:drawableLeft="@drawable/baseline_password_24"
            android:hint="password"
            android:inputType="text"
            android:padding="8sp"
            android:paddingLeft="15sp"
            android:textColor="@color/material_dynamic_neutral30" />
        <Button
            android:id="@+id/loginbuttton"
            android:layout_width="100dp"
            android:layout_height="55dp"
            android:layout_gravity="center"
            android:layout_marginTop="20sp"
            android:gravity="center"
            android:text="Submit" />
      </LinearLayout>
   </androidx.cardview.widget.CardView>
</LinearLayout>
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".BlueTrackUsersDB">

</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_add_notes.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".AddNotes">
   <ImageView
      android:id="@+id/imageView"
      android:layout_width="match_parent"
      android:layout_height="70dp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.0"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.0"
      app:srcCompat="@drawable/nav_bar" />
```

```
    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="39dp"
        android:text="EcoSort"
        android:textSize="32sp"
        android:textStyle="bold"
        android:textAlignment="center"
        android:textColor="@color/black"
        app:layout_constraintBottom_toBottomOf="@+id/imageView"
        app:layout_constraintEnd_toEndOf="@+id/imageView"
        app:layout_constraintHorizontal_bias="0.46"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/imageView" />
    <EditText
        android:id="@+id/notes"
        android:layout_width="336dp"
        android:layout_height="376dp"
        android:layout_gravity="center_horizontal"
        android:background="@drawable/txtbox"
        android:ems="10"
        android:hint="We are glad for adding ur Experience!!!"
        android:inputType="text"
        android:textAlignment="center"
        android:textSize="14sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.494"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.165" />
    <Button
        android:id="@+id/addnote"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Add Description"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.841"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/notes"
        app:layout_constraintVertical_bias="0.136" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
    tools:context=".Home">
    <ImageView
        android:id="@+id/navbar"
        android:layout_width="match_parent"
        android:layout_height="70dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:srcCompat="@drawable/nav_bar" />
    <TextView
        android:id="@+id/title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="EcoSort"
        android:textSize="32sp"
        android:textStyle="bold"
        android:textAlignment="center"
        android:textColor="@color/black"
        app:layout_constraintTop_toTopOf="@+id/navbar"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:padding="10dp"
        app:layout_constraintTop_toBottomOf="@+id/navbar"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/logoutButton" />
    <Button
        android:id="@+id/logoutButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Logout"
        android:textColor="@color/white"
        android:backgroundTint="@color/black"
        android:padding="10dp"
        android:textSize="16sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:id="@+id/loginpage"
    android:layout_height="match_parent"
    android:background="@drawable/img"
    android:gravity="center"
    android:orientation="vertical"
```

```
android:padding="20dp">
<TextView
   android:id="@+id/popup"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:clickable="true"
   android:contextClickable="true"
   android:onClick="showactivistlogin"
   android:text="Activist"
   android:textAlignment="center"
   android:textSize="24sp" />
<androidx.cardview.widget.CardView
   android:layout_width="365dp"
   android:layout_height="318dp"
   app:cardCornerRadius="30dp">
   <ImageView
      android:id="@+id/imageView2"
      android:layout_width="match_parent"
      android:layout_height="67dp"
      app:srcCompat="@drawable/topbox" />
   <TextView
      android:id="@+id/logintxt"
      android:layout_width="match_parent"
      android:layout_height="65dp"
      android:layout_marginTop="13sp"
      android:padding="8dp"
      android:text="LOGIN"
      android:clickable="true"
      android:onClick="showactivistlogin"
      android:textAlignment="center"
      android:textColor="@color/black"
      android:textSize="24sp"
      android:textStyle="bold" />
   <LinearLayout
      android:layout_width="363dp"
      android:layout_height="324dp"
      android:layout_gravity="center_horizontal"
      android:clickable="true"
      android:orientation="vertical">
      <EditText
         android:id="@+id/username"
         android:layout_width="270dp"
         android:layout_height="55dp"
         android:layout_gravity="center"
         android:layout_marginTop="90sp"
         android:background="@drawable/txtbox"
         android:drawableLeft="@drawable/baseline_email_24"
         android:hint="g-mail"
         android:inputType="text"
         android:padding="8sp"
         android:paddingLeft="15sp"
         android:textColor="@color/material_dynamic_neutral30" />
      <EditText
         android:id="@+id/userpassword"
```

```
            android:layout_width="270dp"
            android:layout_height="55dp"
            android:layout_gravity="center"
            android:layout_marginTop="15sp"
            android:background="@drawable/txtbox"
            android:drawableLeft="@drawable/baseline_password_24"
            android:hint="password"
            android:inputType="text"
            android:padding="8sp"
            android:paddingLeft="15sp"
            android:textColor="@color/material_dynamic_neutral30" />
        <Button
            android:id="@+id/userlogin"
            android:layout_width="100dp"
            android:layout_height="55dp"
            android:layout_gravity="center"
            android:layout_marginTop="20sp"
            android:gravity="center"
            android:text="Submit" />
      </LinearLayout>
   </androidx.cardview.widget.CardView>
</LinearLayout>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity"
   android:background="@drawable/img_1">
   <TextView
      android:id="@+id/textView"
      android:layout_width="319dp"
      android:layout_height="46dp"
      android:clickable="false"
      android:text="Welcome To ECOSORT"
      android:textColor="@color/material_dynamic_neutral30"
      android:textSize="30sp"
      android:textStyle="bold"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.467"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.269" />
   <TextView
      android:id="@+id/registertext"
      android:layout_width="98dp"
      android:layout_height="44dp"
      android:text="Register/"
```

```
        android:textColor="@color/material_dynamic_neutral_variant80"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.401"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.575" />
    <TextView
        android:id="@+id/logintext"
        android:layout_width="98dp"
        android:layout_height="44dp"
        android:text="Login"
        android:textColor="@color/material_dynamic_neutral_variant80"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.715"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.574" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_register.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/img_2"
    tools:context=".RegisterActivity">
    <androidx.cardview.widget.CardView
        android:layout_width="317dp"
        android:layout_height="578dp"
        android:foregroundGravity="center"
        app:cardCornerRadius="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.379">
        <ImageView
            android:id="@+id/imageView"
            android:layout_width="match_parent"
            android:layout_height="65dp"
            app:srcCompat="@drawable/topbox" />
        <TextView
            android:id="@+id/registertxt"
            android:layout_width="match_parent"
            android:layout_height="65dp"
            android:textStyle="bold"
```

```
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="24sp"
    android:layout_marginTop="13sp"
    android:text="REGISTER" />
  <LinearLayout
    android:layout_width="363dp"
    android:layout_height="match_parent"
    android:layout_gravity="center_horizontal"
    android:orientation="vertical">
    <EditText
      android:id="@+id/registerusername"
      android:layout_width="270dp"
      android:layout_height="55dp"
      android:layout_gravity="center"
      android:layout_marginTop="90sp"
      android:background="@drawable/txtbox"
      android:drawableLeft="@drawable/baseline_person_24"
      android:hint="username"
      android:inputType="text"
      android:padding="8sp"
      android:paddingLeft="15sp"
      android:textColor="@color/material_dynamic_neutral30" />
    <EditText
      android:id="@+id/registerpassword"
      android:layout_width="270dp"
      android:layout_height="55dp"
      android:layout_gravity="center"
      android:layout_marginTop="15sp"
      android:background="@drawable/txtbox"
      android:drawableLeft="@drawable/baseline_password_24"
      android:hint="password"
      android:inputType="text"
      android:padding="8sp"
      android:paddingLeft="15sp"
      android:textColor="@color/material_dynamic_neutral30" />
    <EditText
      android:id="@+id/registeremail"
      android:layout_width="270dp"
      android:layout_height="55dp"
      android:layout_gravity="center"
      android:layout_marginTop="15sp"
      android:background="@drawable/txtbox"
      android:drawableLeft="@drawable/baseline_email_24"
      android:hint="Email"
      android:inputType="textEmailAddress"
      android:padding="8sp"
      android:paddingLeft="15sp"
      android:textColor="@color/material_dynamic_neutral30" />
    <EditText
      android:id="@+id/registerrole"
      android:layout_width="270dp"
      android:layout_height="55dp"
      android:layout_gravity="center"
```

```
            android:layout_marginTop="15sp"
            android:background="@drawable/txtbox"
            android:drawableLeft="@drawable/baseline_person_24"
            android:hint="Role (Activist/User)"
            android:inputType="text"
            android:padding="8sp"
            android:paddingLeft="15sp"
            android:textColor="@color/material_dynamic_neutral30" />
        <EditText
            android:id="@+id/registerqualification"
            android:layout_width="270dp"
            android:layout_height="55dp"
            android:layout_gravity="center"
            android:layout_marginTop="15sp"
            android:background="@drawable/txtbox"
            android:drawableLeft="@drawable/baseline_person_24"
            android:hint="qualification"
            android:inputType="text"
            android:padding="8sp"
            android:paddingLeft="15sp"
            android:textColor="@color/material_dynamic_neutral30" />
        <Button
            android:id="@+id/registerbutton"
            android:layout_width="126dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="20sp"
            android:gravity="center"
            android:text="Register" />
    </LinearLayout>
  </androidx.cardview.widget.CardView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_note.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:layout_margin="10dp"
  app:cardCornerRadius="12dp"
  app:cardElevation="5dp">
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">
    <TextView
      android:id="@+id/note_series_number"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="1"
      android:textStyle="bold"
      android:textSize="18sp"
```

```
            android:textColor="@android:color/black"/>
        <TextView
            android:id="@+id/note_content"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:text="Article Content"
            android:textSize="16sp"
            android:textColor="@android:color/darker_gray"/>
    </LinearLayout>
</androidx.cardview.widget.CardView>
```

7. RESULT/OUTPUT SCREEN SHOTS

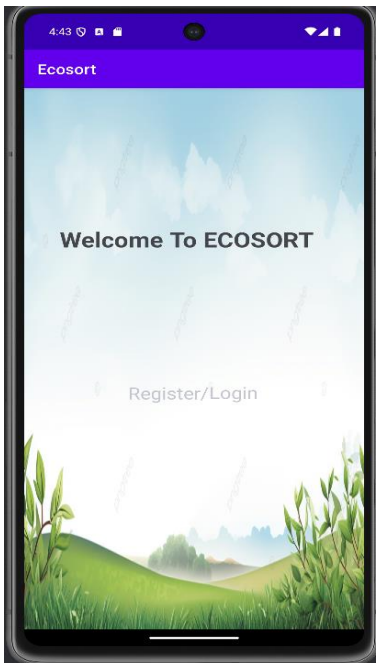


Fig-1

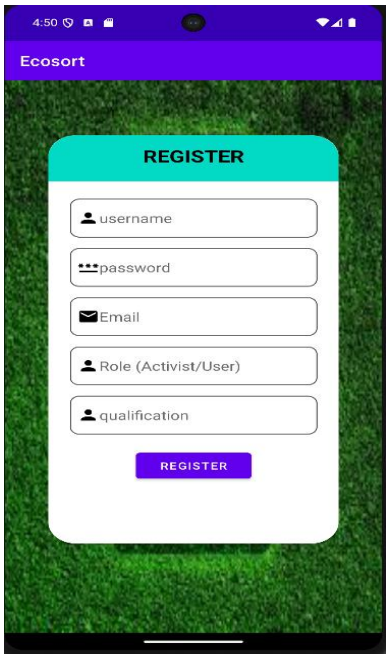


Fig-2

Figure 1: Welcome Page – The Welcome Page of Ecosort serves as the initial screen, greeting users with a visually appealing interface. It provides an option to register or log in.

Figure 2: Register Page – The Register Page of Ecosort allows new users to create an account by entering their username, password, email, role (Activist/User), and qualification. The intuitive form ensures a smooth onboarding process. The vibrant design aligns with the app's eco-friendly theme.

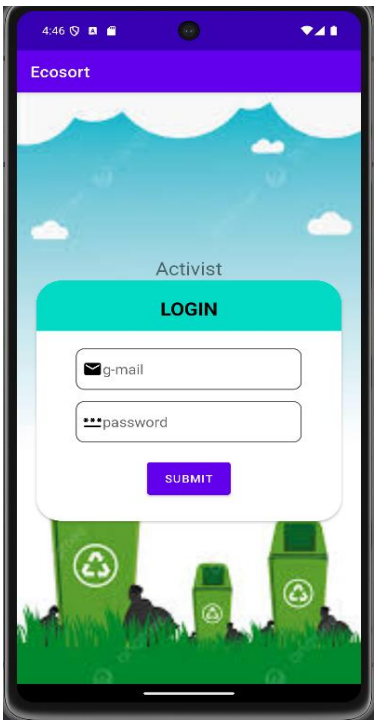


Fig- 3

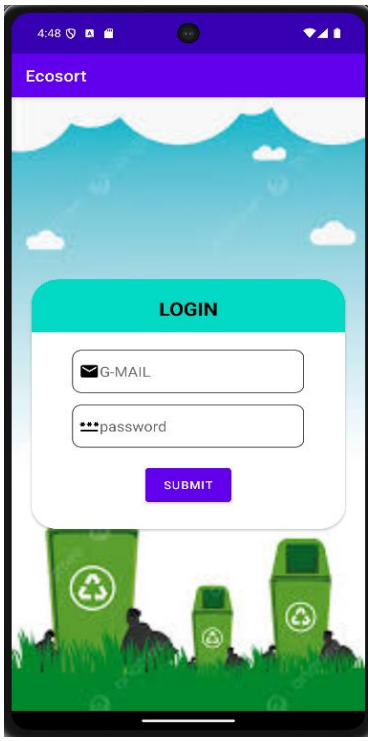


Fig- 4

Figure 3*: *Normal User Login Page – This is the standard login interface for regular users. It allows them to enter their **email and password** to access the app. If they are activists, they can click on the **"Activist"** option to navigate to the **Activist Login Page** for role-specific access.

Figure 4: Activist Login Page – This login interface is specifically for activists. Users must enter their email and password to access activist-related features. Unlike the normal user login, this page is designed for users engaging in environmental activism within the Ecosort platform

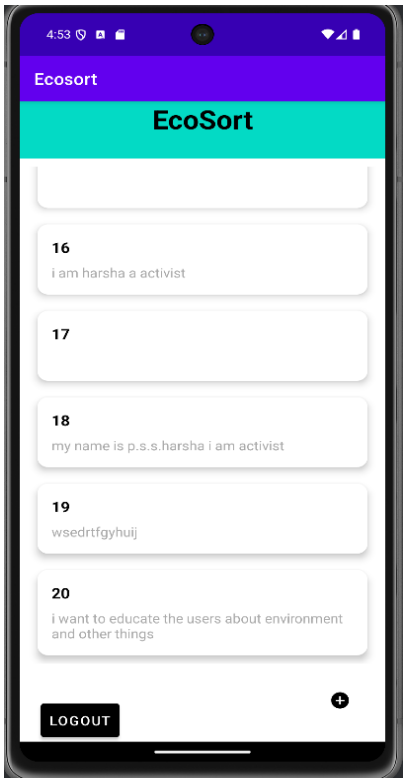


Fig- 5

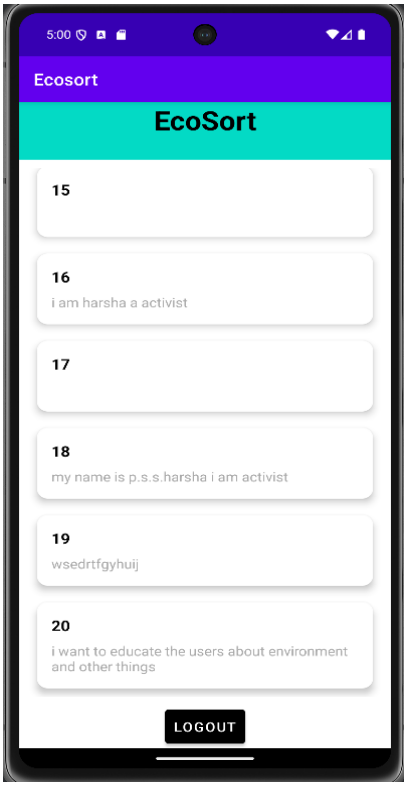


Fig- 6

Figure 5: Activist Home Page – This page serves as the central hub for activists in the Ecosort platform. It displays a list of descriptions or posts added by activists, aiming to spread awareness about environmental issues. Activists can add new posts and log out from this screen.

Figure 6: Activist Dashboard – This page displays a list of activist contributions, where users can view shared experiences and environmental insights. It serves as a platform for activists to educate and inspire others. A Logout button allows users to exit their session securely.

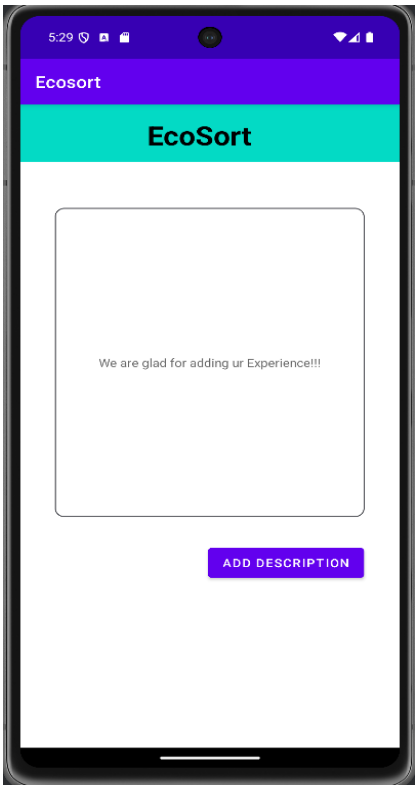


Fig- 7

Figure 7: Add Description Page – This page allows activists to contribute their experiences and insights by adding descriptions. Users can type their message in the provided text box and submit it using the "**Add Description**" button, helping to spread awareness about environmental issues.

8. CONCLUSION

In conclusion, **Ecosort** emerges as an innovative and essential platform for promoting environmental awareness and sustainability. By leveraging modern technology such as Android, Java, and Firebase, the application empowers users—both activists and general users—to share, educate, and engage in eco-friendly practices effectively. The integration of user-generated content and community-driven insights fosters awareness and action, helping individuals contribute to a greener planet.

The application's modular design and user-friendly interface ensure accessibility for people from all demographics. From sharing environmental experiences and educational content to engaging in discussions on sustainability, **Ecosort** provides a holistic solution for promoting environmental responsibility. Moreover, its ability to store and display user-generated insights encourages collaboration and collective efforts toward positive ecological impact.

As the platform evolves, **Ecosort** has the potential to integrate advanced features such as AI-based content recommendations, gamification for user engagement, and real-time environmental impact tracking. By building on these future enhancements, the application can significantly expand its influence, enabling individuals and communities to take meaningful action toward a sustainable future. With its mission to foster environmental responsibility, **Ecosort** stands as a timely and impactful contribution to addressing the pressing environmental challenges of our era.

9. REFERENCES (WEB SITE URLS)

GitHub repository link: https://github.com/yaswan123/Ecosort

1. **Firebase Documentation: https://firebase.google.com/docs**
2. **Firebase Realtime Database: https://firebase.google.com/docs/database**
3. **Android Developer Guide: https://developer.android.com/guide**
4. **Android Studio: https://developer.android.com/studio**